

## SUMEX-AIM RESOURCE PROGRESS REPORT - YEAR 06

This annual report covers work performed under NIH Biotechnology Resources Program grant RR-785 supporting the Stanford University Medical EXperimental Computer (SUMEX) research resource for applications of Artificial Intelligence in Medicine (AIM). It spans the year from May 1978 - April 1979.

## 2 Resource Operations

### 2.1 Progress

#### 2.1.1 Resource Summary and Goals

The SUMEX-AIM project is a national computer resource with a dual mission: a) the promotion of applications of computer science research in artificial intelligence (AI) to biological and medical problems and b) the demonstration of computer resource sharing within a national community of health research projects. The SUMEX-AIM resource is located physically in the Stanford University Medical School and is administered jointly under the Stanford Departments of Genetics and Computer Science. SUMEX-AIM serves as a nucleus for a community of medical AI projects at universities around the country. SUMEX provides computing facilities tuned to the needs of AI research and communication tools to facilitate remote access, inter- and intra-group contacts, and the demonstration of developing computer programs to biomedical research collaborators.

#### Overview of AI Research

Artificial Intelligence research is that part of Computer Science concerned with symbol manipulation processes that produce intelligent action (1). By "intelligent action" is meant an act or decision that is goal-oriented, is arrived at by an understandable chain of symbolic analysis and reasoning steps, and utilizes knowledge of the world to inform and guide the reasoning. Some scientists view the performance of complex symbolic reasoning tasks by computer

---

(1) For recent reviews to give some perspective on the current state of AI, see: (i) Boden, M., "Artificial Intelligence and Natural Man," Basic Books, New York, 1977; (ii) Feigenbaum, E.A., "The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering," Proceedings of the Fifth International Conference on Artificial Intelligence, 1977; (iii) Winston, P.H., "Artificial Intelligence", Addison-Wesley Publishing Co., 1977; and (iv) Nilsson, N.J., "Artificial Intelligence", Information Processing 74, North-Holland Pub. Co. (1975). An additional overview of research areas and techniques in AI is being developed as an "Artificial Intelligence Handbook" under Professor E. A. Feigenbaum by computer science students at Stanford (see page 130 for a status report and Appendix I for a current outline).

programs as the sine qua non for artificial intelligence programs, but this is necessarily a limited view.

Another view unifies AI research with the rest of computer science. It is a simplification, but worthy of consideration. The potential uses of computers by people to accomplish tasks can be "one-dimensionalized" into a spectrum representing the nature of the instructions that must be given the computer to do its job; call it the WHAT-TO-HOW spectrum. At the HOW extreme of the spectrum, the user supplies his intelligence to instruct the machine precisely HOW to do his job, step-by-step. Progress in computer science may be seen as steps away from that extreme "HOW" point on the spectrum: the familiar panoply of assembly languages, subroutine libraries, compilers, extensible languages, etc. illustrate this trend.

At the other extreme of the spectrum, the user describes WHAT he wishes the computer to do for him to solve a problem. He wants to communicate WHAT is to be done without having to lay out in detail all necessary subgoals for adequate performance yet with a reasonable assurance that he is addressing an intelligent agent that is using knowledge of his world to understand his intent, complain or fill in his vagueness, make specific his abstractions, correct his errors, discover appropriate subgoals, and ultimately translate WHAT he wants done into detailed processing steps that define HOW it shall be done by a real computer. The user wants to provide this specification of WHAT to do in a language that is comfortable to him and the problem domain (perhaps English) and via communication modes that are convenient for him (including perhaps speech or pictures).

The research activity aimed at creating computer programs that act as "intelligent agents" near the WHAT end of the WHAT-TO-HOW spectrum can be viewed as the long-range goal of AI research. Historically, AI research has been the primary vehicle for progress toward this objective, although a substantial part of the applied side of computer research and development has related goals, if an often fragmented approach. Unfortunately, workers in other scientific disciplines are generally unaware of the role, the goals, and the progress in AI research.

Currently authorized projects in the SUMEX community are concerned in some way with the design of "intelligent agents" applied to biomedical research. The tangible objective of this approach is the development of computer programs that, using formal and informal knowledge bases together with mechanized hypothesis formation and problem solving procedures, will be more general and effective consultative tools for the clinician and medical scientist. The systematic search potential of computerized hypothesis formation and knowledge base utilization, constrained where appropriate by heuristic rules, empirical data, or interactions with the user, has already produced promising results in areas such as chemical structure elucidation and synthesis, diagnostic consultation, and modeling of psychological processes. Needless to say, much is yet to be learned in the process of fashioning a coherent scientific discipline out of the assemblage of personal intuitions, mathematical procedures, and emerging theoretical structure of the "analysis of analysis" and of problem solving. State-of-the-art programs are far more narrowly specialized and inflexible than the corresponding aspects of human intelligence they emulate; however, in special domains they may be of comparable or greater power, e.g., in the solution of formal problems in organic chemistry or in the integral calculus.

### Resource Sharing Goals

An equally important function of the SUMEX-AIM resource is an exploration of the use of computer communications as a means for interactions and sharing between geographically remote research groups engaged in biomedical computer science research. This facet of scientific interaction is becoming increasingly important with the explosion of complex information sources and the regional specialization of groups and facilities that might be shared by remote researchers (2). Our community building role is based upon the current state of computer communications technology. While far from perfected, these developing capabilities offer highly desirable latitude for collaborative linkages, both within a given research project and among them. Several of the active projects on SUMEX are based upon the collaboration of computer and medical scientists at geographically separate institutions; separate both from each other and from the computer resource. The network experiment also enables diverse projects to interact more directly and to facilitate selective demonstrations of available programs to physicians, scientists, and students. Even in their current developing state, communication facilities enable effective access to the rather specialized SUMEX computing environment from a great many areas of the United States (and to a more limited extent from Canada, Europe, and other international locations). In a similar way, the network connections have made possible close collaborations in the development and maintenance of system software with other facilities.

### Synopsis of Last Year's Progress

As we complete year 06, the first year of our recent 3-year continuation grant, we can report substantial further progress in the overall mission of the SUMEX-AIM resource. We have continued the refinement of an effective set of hardware and software tools to support the development of large, complex AI programs for medical research and to facilitate communications and interactions between user groups. We have worked to maintain high scientific standards and AI relevance for projects using the SUMEX-AIM resource and have actively sought new applications areas and projects for the community. Many projects are built around the communications network facilities we have assembled; bringing together medical and computer science collaborators from remote institutions and making their research programs available to still other remote users. As discussed in the sections describing the individual projects, a number of the computer programs under development by these groups are maturing into tools increasingly useful to the respective research communities. The demand for production-level use of these programs has surpassed the capacity of the present SUMEX facility and we have been investigating the general issues of how such software systems can be moved from SUMEX and supported in production environments.

---

(2) A recent perspective on the scientific and financial aspects of technological resource sharing can be found in Coulter, C. L., Research Instrument Sharing, Science, Vol. 201, No. 4354, August 4, 1978.

A number of significant events and accomplishments affecting the SUMEX-AIM resource occurred during the past year:

- 1) On July 1, 1978, Professor Edward Feigenbaum, chairman of the Stanford Department of Computer Science, assumed the role of SUMEX Principal Investigator following Professor Joshua Lederberg's installation as president of The Rockefeller University. We have smoothly completed the management transition and the SUMEX-AIM project and community continue to operate with the same high level of vitality. Professor Lederberg continues to maintain close ties with SUMEX activities as chairman of the SUMEX-AIM Executive Committee. Professor Stanley Cohen, Dr. Lederberg's successor as chairman of the Stanford Department of Genetics, assists in the coordination of project activities with medical research.
- 2) We have continued development of the SUMEX facility hardware and software systems to enhance throughput and to better control the allocation of resources. We also completed installation and evaluation of a connection to TELENET as an alternate source of communications services for our community.
- 3) A first version of the AGE system, partially supported under the SUMEX core research effort, has been completed. It uses the "blackboard model" for coordinating multiple expert sources of knowledge for the solution of problems. This system provides the general control structure and an interactive facility for implementing representations of expert knowledge sources and is being used experimentally by one of the new SUMEX-AIM projects to design a program for modeling aspects of human cognition.
- 4) We successfully completed the design and a demonstration of the MAINSAIL language system as a tool for software portability. A common compiler and code generators and runtime support for TENEX, TOPS-10, TOPS-20, RT-11, RSX-11, and UNIX have been developed as part of this demonstration system and numerous applications programs written by collaborating research groups. Further work past this demonstration phase will be done independently of SUMEX through a private company being formed to continue the development, dissemination, and maintenance of MAINSAIL.
- 5) We have completed plans for a satellite machine that will be able to support more operational demonstrations of mature AI programs and help alleviate system congestion for on-going program development. A proposal for acquiring a DEC 2020 system meeting our requirements is pending approval by the NIH-BRP. We have also assisted the DENDRAL project in planning an independent system suitable for further development and export of chemical structure elucidation programs into the biochemical community.
- 5) The progress of SUMEX-AIM user projects in the development of their respective programs is reported by the individual investigators. We have worked hard to meet their needs and are grateful for their expressed appreciation.

### 2.1.2 Technical Progress

The following material covers SUMEX-AIM resource activities over the past year in greater detail. These sections outline accomplishments in the context of the resource staff and the resource management. Details of the progress and plans for our external collaborator projects are presented in Section 4 beginning on page 64.

#### 2.1.2.1 Facility Hardware Development

Over the past year, the SUMEX KI-10 configuration, shown in Figure 1, has changed little and continues to operate effectively within its capacity limitations. We completed the procurement of the Systems Concepts SA-10 channel adapter including all parts outstanding as of the last report. This subsystem, with the Calcomp disks and tapes, has functioned very reliably over the past year.

Our primary new facility hardware development efforts this year have been directed at:

- 1) Selection of a satellite processor to allow more operational demonstrations of mature AI programs and to ease loading congestion.
- 2) Planning for the integration of the satellite machine into the KI-10 facility.
- 3) Implementing local communication line control facilities to make more efficient use of available scanner ports.

These are discussed in more detail below.

#### Loading Background

The SUMEX-AIM facility has been operating at capacity in terms of prime-time computing load for the past several years as documented in our previous annual reports. In spite of implementing a number of strategic facility augmentations over the years, we have not been able to satisfy the computing demands of our community. This condition has constrained the growth of the AIM community and our ability to bring AI programs nearing operational status in contact with potential external user communities while continuing to support on-going program development efforts. We have taken active steps to transfer prime time interactive loading to evening and night hours as much as possible including shifting personnel schedules (particularly for Stanford-based projects). We have also implemented tools to control the fair allocation of CPU resources between various user communities and projects and have encouraged jobs not requiring intimate user interaction to run during off hours using batch job facilities. Despite these efforts, our prime time loading has remained at saturation. Perhaps the most significant effect of the resulting poor response time is the deterrence of interactions with medical and other professional collaborators experimenting with available AI programs, whose schedules cannot be adjusted to

meet computer loading patterns. This has hampered the more extensive testing of mature programs such as INTERNIST, MYCIN, CONGEN, SECS, and PUFF.

This continuing saturation brought about serious discussion about the scope of computing needs of the AIM community and possible justification of additional PDP-10 scale machines to be added to the AIM network. Several specific proposals were submitted for additional user nodes. Only one of those has been approved to date, for a DEC 2050 system at Rutgers University which was brought on-line late in the summer of 1978. A small part of that machine's capacity is available now to support AIM community needs outside of Rutgers.

From the SUMEX viewpoint, we have attempted to do everything feasible and economically justified within available budgets to maximize the use of the existing hardware for productive work. We have effectively exhausted available avenues for augmenting the current KI-10 machines. Some advantage would be gained by additional core memory but we do not feel the improvement would be sufficient to justify the investment at this time. An upgrade to a more capable KI-10 system is beyond our budget limitations and may be premature in any case in light of projected developments in new machine architectures outlined in Appendix II.

As discussed in our renewal application for this grant term, an alternative approach to meet community computing needs is to explore the use of smaller, less expensive machines as satellites to the KI-TENEX system. Such systems have been under active development during recent years and could have several advantages including:

- 1) A relatively small investment in capital equipment is required for each incremental augmentation.
- 2) Possible closer location to individual research groups thereby allowing better human engineering of user interfaces by using higher speed communication lines and display technology.
- 3) An improved allocation flexibility by having to satisfy fewer simultaneous scheduling constraints and by being more easily dedicatable to operational demonstrations.

One disadvantage of this approach is that each such machine would have a lower capacity and it would be difficult to aggregate such dispersed capacity when needed for a single computing-intensive task. This suggests the continuing need for a spectrum of machine configurations from small "personalized" machines to large centralized resources. Nevertheless, we feel the capacity of available small machines is sufficient to support several simultaneous users and warrants serious consideration as both a means for incrementally augmenting the SUMEX resource and for dispersing computing power as justified to individual user groups.

Based on the Council approval of this approach in our renewal application, our plans for acquiring such a satellite machine and for integrating it into the KI-10 system with a local network are described below.

It should also be noted that we have encouraged projects with specific needs for more operational demonstration or export of programs to consider acquiring their own machines in order to preserve SUMEX resources for new program development and for support of projects unable to justify their own machine currently. The DENDRAL project has proposed a VAX machine for such a purpose that would be integrated into the SUMEX facility but dedicated to support of the DENDRAL biomolecular characterization community. The choice of VAX was made to provide the best match with machines increasingly available in a biochemistry laboratory environment and able to run the programs being developed by DENDRAL (including CONGEN recently converted from INTERLISP to BCPL). At the same time the choice of VAX is advantageous to SUMEX in that it would give us experience with that machine in line with current projections that VAX will become the "standard" DEC computing product and that the ARPANET AI community will implement a VAX INTERLISP system (see Appendix II).

#### Satellite Machine Selection

Over the past year we have spent considerable effort evaluating strategies and alternatives for implementing the planned satellite machine. The key requirement for any such machine to meet pressing community needs is that it be software-compatible with the existing INTERLISP and basic monitor functions available on the SUMEX KI-10 systems and the Rutgers DEC-2050. This will allow programs, written for the most part in INTERLISP, to move easily from development stages to demonstration trials and back with a minimum of reprogramming. A second requirement is that the system be inexpensive in order to minimize initial capital outlay and to allow other groups to purchase similar systems for their own needs.

As detailed in Appendix II, we have been in a period of transition in computing technology. More compact and inexpensive yet powerful machines have become available and new directions in machine architecture are being adopted emphasizing large address spaces and improved instruction sets for user program support. In several years, we expect the PDP-10/20 architecture to begin to be replaced by larger address space and more cost-effective systems (most likely VAX). We do not expect even early versions of these new systems that support INTERLISP to be available for at least two years, however. Thus, in order to meet the immediate needs of the SUMEX-AIM community, we feel the best approach is to acquire a PDP-10-compatible system as soon as possible.

There are two alternative systems available that meet our requirements for a satellite machine within budget limitations; the DEC 2020 and the Foonly F2. We have evaluated both of these candidate machines (see Appendix II) and have run benchmarks on the 2020 (the only one of the two machines with fully working system in the field). These data, shown in Figure 3, compare 2020 responsiveness under load against single- and dual-processor KI-10 systems. As can be seen, the 2020 is a bit more than half the speed of a single KI-10 and can be expected to support up to three active LISP users simultaneously. This upper bound is limited principally by page swapping capacity. Based on published specifications, we expect the Foonly F2 would perform comparably.

We feel that the DEC 2020 is the more advantageous solution. A used 2020 is deliverable almost immediately at a major discount from list price (pricing details have been submitted separately). It is known to be reliable, runs a

monitor compatible with INTERLISP and the most current DEC software, and will be maintainable by DEC for many years. It will also likely retain a better resale value in future years. Whereas the F2 is potentially more cost-effective (its quoted purchase price is below that of the discounted 2020), it has a highly uncertain delivery schedule and no performance track record. It also has no assurance of routine maintenance, vendor support, or resale value. In the long term we feel these uncertainties and the extra in-house effort that would be required to maintain and support the F2 offset its initial price advantage. Thus, the DEC 2020 is the better choice to provide an immediate, effective, and reliable solution to SUMEX-AIM community computing needs.

Based on benchmark performance and needs for integrating a 2020 system into the SUMEX facility, we have proposed the following configuration for the machine:

- 2020 Processor and console
- 512K words of memory
- 1 200 Mbyte disk drive (RP-06)
- 16 asynchronous communication lines
- TU-45 tape drive
- TOPS-20 software

A proposal is pending with NIH/BRP to approve purchase of this machine.

#### Satellite Machine Integration

The introduction of satellite machines into the SUMEX facility raises important issues about how best to integrate such systems with the existing machines. We seek to minimize duplication of peripheral equipment and interdependence among machines that would increase failure modes. We also require high-speed intermachine file transfer capabilities and terminal access arrangements allowing a user to connect flexibly to any machine of choice in the resource.

The initial design of the SUMEX system was that of a "star" topology centered on the KI-10 processors. In this configuration, all peripheral equipment and terminal ports were connected directly to the KI-10 busses. With the addition of a satellite machine, a unique focus no longer exists and some pieces of equipment need to be able to "connect" to more than one host. For example, a user coming into SUMEX over TYMNET will want to be able to make a selection of which machine he connects to. Another TYMNET user may want to make another choice of machine and so the TYMNET interface needs to be able to connect to any of the hosts. This could be accomplished by creating separate interfaces for each of the hosts to the TYMNET, each with a different address. Besides being expensive to duplicate such interfaces, it would be inconvenient for a user to reconnect his terminal from one host to another. He would have to break his existing connection and go through another connect/login process to get to another machine. Since we want to facilitate user movement between various machines in the SUMEX resource, this process needs to be as simple as possible - in fact a user may have jobs running simultaneously on more than one machine at a time.

Similarly, we need to be able to quickly transfer files between any two machines in the resource, connect common peripheral devices (e.g. printer or

plotter) to any machine desiring to use them, and allow any host to access other remote resources such as Stanford campus printers or terminal clusters. If we were to establish direct connections pairwise between machines and devices, the number of such connections would go up quadratically with the number of devices.

A more effective solution lies in the implementation of a local network in which all devices (host CPU's, peripheral devices, network gateways, etc.) are tied to a common communications medium and can thereby establish logical connections as needed between any pair of nodes. Such network systems have been under development for a number of years, taking on various topological configurations and control structures depending on bandwidth requirements and interdevice distances. A very attractive design for a highly localized system configuration from the viewpoint of simplicity, reliability, and bandwidth is the Ethernet which has been under development for several years at Xerox Palo Alto Research Center (3). The simplest form of Ethernet interconnection for a facility like SUMEX would be a single bus shared by all devices (see Figure 2). The Ethernet utilizes a fully distributed control structure in that each device connected to the net can independently decide to send a message to any other device on the net depending on the functions it is actively performing. Of course, decisions about which devices need to communicate with each other at a given time and what the precise message content is are determined by higher level system activities and requests, for example to implement a file transfer, mail forwarding, teletype connection, printer output, etc. As long as the net is not in use and only one device at a time is attempting to transmit, no problem occurs. The sending device transmits its packet of information which contains a destination address, packet type designator, and error detection codes. All other devices on the net continuously "listen" to what is being sent and the one assigned the appropriate destination address picks up the packet, acknowledges its receipt, and processes it. If the packet address is garbled by errors or no device with the appropriate address exists, the sender "times-out" and decides how to proceed based on the higher level function being performed. Packets are kept short relative to network bandwidth so that a given device cannot "hog" the net.

However, if two or more devices decide to transmit over the shared medium at the same time, a "collision" occurs and a mechanism must exist to detect the collision and to select one of the contending devices to go first. Since this contention arbitration is the fundamental characteristic of the control structure of such nets, they are commonly called "contention" networks. In the Ethernet, a collision is detected by each sending device listening to what is being transmitted on the bus. If a transmission is already in progress, the device waits until the net is quiet for a period before starting to send. When it does transmit, it continues to listen to what is going over the communications line and compares that data with what it is sending. If a disagreement is detected the device assumes that some other device has started to transmit at the same time and aborts its transmission. A time window exists between the start of a transmission and when all devices can be assumed to know that a transmission is in progress. This interval is given by the speed of the net and the distance between the sending node and its most distant neighbor. If a collision is detected, the net is "jammed" with noise for a period such that all devices know

---

(3) See Metcalfe, R. M. and Boggs, D. R., "Ethernet: Distributed Packet Switching for Local Computer Networks," Comm. ACM, Vol. 19, No. 7, July 1976.

a collision has occurred and then each sending device waits a random period of time to begin retransmission. This random delay is what sequences devices so that a deadlock of successive collisions is avoided (4).

More complex networks can be created with several Ethernets by having one of the nodes on the network be a "gateway" that knows how to communicate with another Ethernet or some other external network. These gateways can translate between packet conventions used in the Ethernet and those used in the ARPANET, TYMNET, TELENET, etc.

Xerox has implemented internally an extensive set of Ethernets with interconnections between them and with other external networks. These local networks operate at 5-10 Mbits/sec over distances of about 1 kilometer and perform well in terms of efficient use of the transmission medium and low latency between deciding to transmit and being able to get access to the medium (5). The Stanford Computer Science Department will be one of three recipients of grants from Xerox that will include Ethernet connection hardware. Since the Computer Science Department systems are integrally connected with a major user group on SUMEX (the Heuristic Programming Project) and since the Ethernet design is ideal for the integration of new satellite machines with the existing SUMEX facility, we have chosen it as the model for our planned facility changes. The proposed new topological design is shown in Figure 2 and will include creating new interfaces for each host machine, the TYMNET, the local teletype scanner, other peripheral devices, and a gateway to other local networks (e.g., the Computer Science Department machine and planned terminal clusters).

#### Communications Hardware Development

A final area of hardware development concerns communications. We have implemented line disconnect control hardware on local telephone lines similar to what exists logically for our network connections. Previously we were unable to detect when carrier dropped on phone connections, for example when a user hung up without logging out or was accidentally disconnected during a session. This left his job hanging so that the next person dialing up in that line would automatically be connected to the earlier job resulting in possible privacy or security loss. The system now receives a hardware interrupt when a line drops and if the job that was on that line is still active, the job is detached so it can be picked up and continued. Conversely, when a user logs out, we do an automatic disconnect on his phone line so that our incoming rotaries are not congested with unused, hoarded phone connections.

---

(4) A similar type of local network called CHAOSNET has been under development at MIT. It differs from Ethernet in that it uses delay counters to sequence colliding devices. The delay for each sender is determined by counting down at a prespecified rate the arithmetic difference in node address between the last successful transmission and the prospective sender. Thus by selecting node addresses corresponding roughly to the physical position of a node on the net, proper interleaving can be achieved to arbitrate collisions.

(5) See Shoch, J. F. and Hupp, J. A., "Performance of an Ethernet Local Network -- A Preliminary Report," Proceedings of the Local Area Communications Network Symposium, Boston, May 1979.

We are also developing a switch to allow more effective use of the 64 available teletype scanner ports. We typically have about 40-50 jobs on the system during peak loads (mid-afternoon) of which 10 are detached, 10 come from network or pseudo-teletype connections, 10 come from local dialup connections, and 15 come from leased or hard-line connections. With this mix the 64 scanner ports on the system are adequate. However, high speed displays or leased lines require dedicated ports whether or not they are in use and thus the scanner is overloaded with fixed line assignments, many of which are not in simultaneous use. We have looked at the economics of adding another scanner or of making it possible to switch available scanner ports to active lines and the switch is the more cost-effective. A microprocessor-based switch is now being installed and tested that will allow us to selectively connect 32 scanner ports to any of 64 dedicated lines.

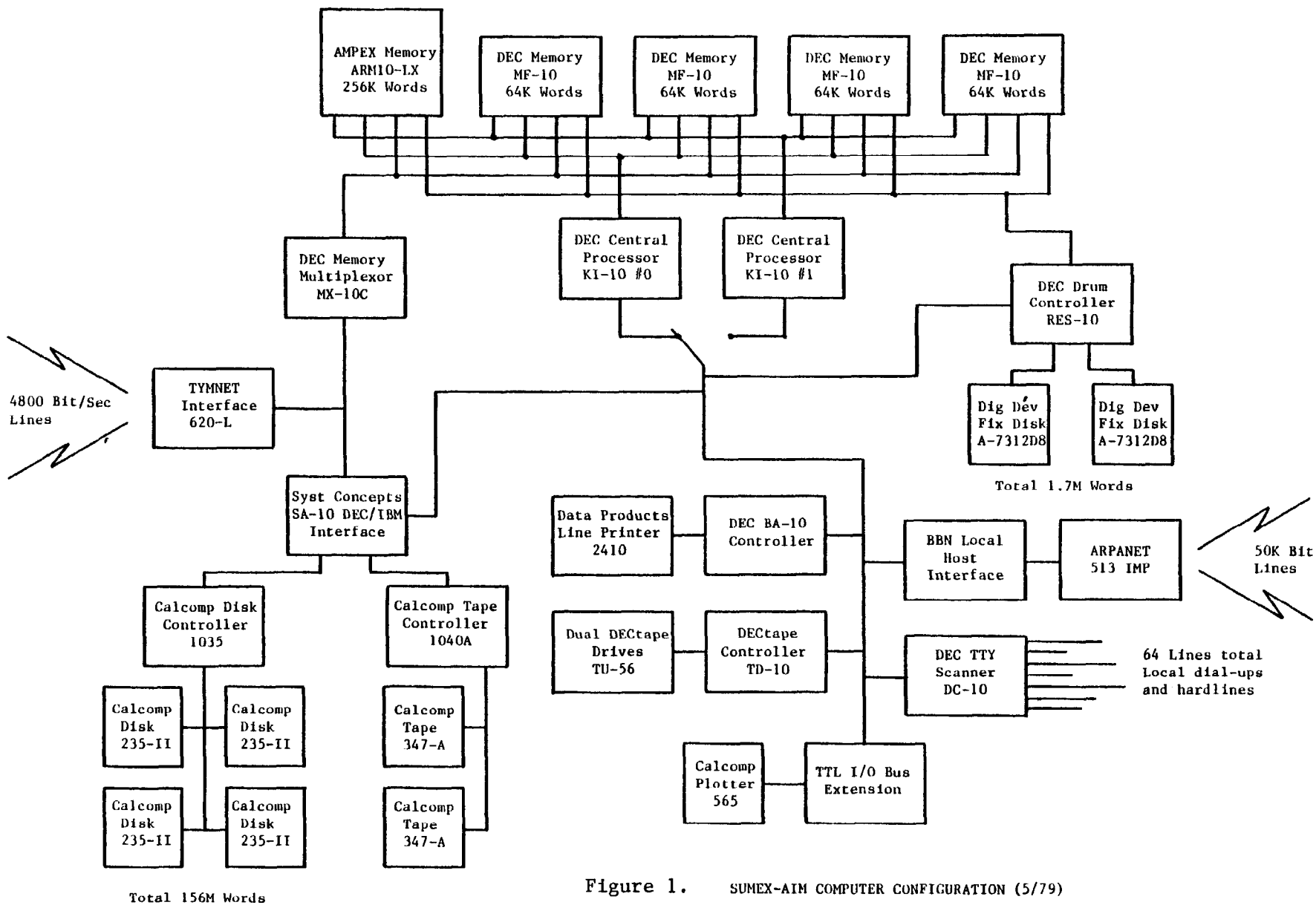


Figure 1. SUMEX-AIM COMPUTER CONFIGURATION (5/79)

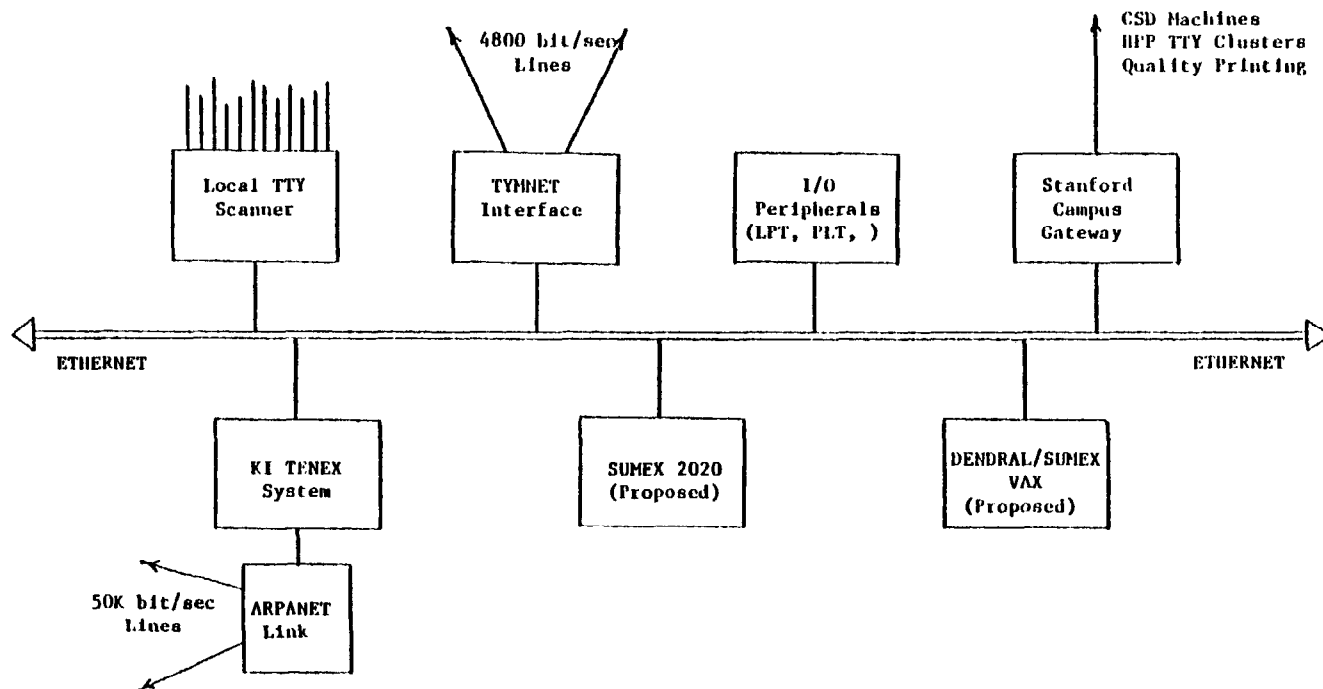


Figure 2. Planned Intermachine Connections via ETHERNET

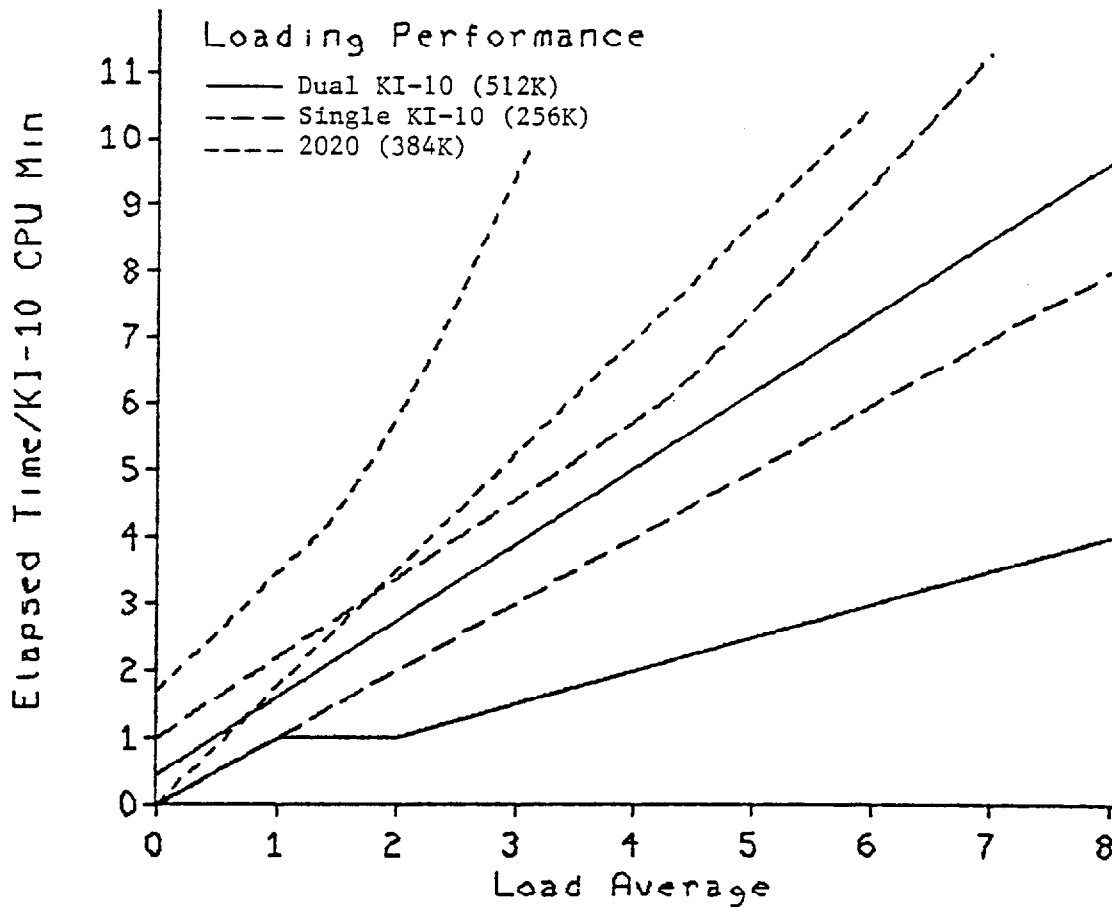


Figure 3. DEC KI-10 Versus 2020 Performance Under Load. For each of the three machine configurations, two graphs are given. The lower graph shows performance for small, CPU-intensive jobs and the upper graph shows performance for large, page-fault-intensive jobs. These curves bound the expected performance for typical user jobs. It is assumed that a KI-10 averages about 1.7 times the speed of a 2020.

### 2.1.2.2 System Software Development

Our system software work this past year has concentrated on several areas including system changes reflecting hardware development projects, correcting various system bugs, improved community loading controls, and implementing new features for better user community support.

#### Hardware Implementation

System work was required to enable the installation of the TELENET equipment (see Section 2.1.2.3) and the local communication line control hardware. We implemented "Xon/Xoff" facilities for the TELENET interface so that all terminals could run at an effective 1200 baud rate with output flow controlled by appropriate network "backpressure" commands when buffers fill for slower terminals. These changes were completed in the fall when the final evaluation of TELENET took place and significantly smoothed network output flow over what had been available before.

Servers were also implemented to handle the interrupt and I/O bus interfaces for the line disconnect control hardware and for the hardline switch interface. The switch interface is still in the process of being debugged.

#### Monitor Bug Fixes and Improvements

We found a number of subtle bugs in the system this past year that had been causing periodic problems in hung jobs or crashes. By now, all of the "obvious" bugs have been located and so those remaining are much more elusive, occurring infrequently or only after a long chain of rare events that is difficult to reconstruct. Examples of fixes include problems in DDMP, the program that periodically migrates altered file pages from core to refresh the disk image of these pages. Two bugs existed, one that caused infrequent error logging calls to mishandle the stack and one that overlooked certain pages under the assumption that future core garbage collections would take care of them. This latter bug caused relatively frequent file errors during crashes or when taking the system down because the overlooked pages were never refreshed on disk by core garbage collection since the system halted. We have had a significantly more reliable file system during crashes as a result of this fix.

Several bug fixes were made in the ARPANET code having to do with the handling of special control packets when aborting partially created connections and the release of connections after transmission errors had occurred.

We also found a bug in the fork manipulation code that caused jobs to hang occasionally when multiple fork manipulations were going on simultaneously. These resulted when two forks were attempting to examine the job fork structure data base, one got interrupted in progress, and the other made some changes that altered information in the tables that the first fork expected to remain as set up when it was interrupted.

A number of additional improvements were made to upgrade various monitor routines and JSYS's to conform with TENEX 1.34, to checksum monitor code as loaded to detect I/O errors or memory problems, to make the console teletype of the second processor available for use, and to improve operational procedures for taking crash dumps and reloading the system.

System Loading Controls

We previously reported on the system load controls we have implemented to allocate available system capacity effectively among projects and users according to Executive Committee guidelines. These include:

- 1) A "soft" CPU percentage control, assisted by a program which adjusts user percentages for the scheduler based on the dynamic loading of the system. This allocation control structure uses the scheduler's five queue system that ranks processes according to their degree of interactiveness (CPU time between requests for teletype inputs). Processes in the highly interactive queues (text editing, etc.) are scheduled at highest priority without consideration of allocation percentages. If no processes are runnable from these queues, more CPU-bound queues are scanned and processes are selected for running based on how much of their allocated time has been consumed during a given allocation control cycle time (currently 100 seconds). This system is not a reservation system in that it does not guarantee a given user some percentage of the system. It allocates cycles preferentially, trading off a priori allocations with actual demand but does not waste cycles.
- 2) an overload control mechanism that operates during peak loading periods to limit the number of active processes on the system to those that can be reasonably supported with acceptable response time. This avoids slaving all users to their terminals waiting inefficiently for the machine cycles they need to get useful work done when there are not enough to go around. Each project receives a pro rata share of the active slots the system can accommodate. Rather than allow many users to vie unproductively for each project's slots (as in a pie-slice system), we ask selected users within each group to restrict their use for periods of 20 minutes so that those remaining can work effectively within the project aliquot. Allocation of active slots is made on the basis of relative community and project percentage allocations (assigned by the AIM Executive committee). Within each project, slots are allocated either on a round-robin basis or taking into account optional project priorities among users. Under overload conditions, active jobs outside of the available slots are asked to slow down, thereby holding the load within tolerable limits. If such jobs do not voluntarily cooperate, they may be forced to comply.

This system has been in operation for the past year and has operated quite well. We continued to place no load limiting controls on the national AIM community projects, however, since they have historically consumed been below their allocated quota. Stanford users and staff have adapted their expectations of system response and find it more productive to coordinate their time on the machine with others in their project so as to work on a more lightly loaded system. Indeed, as can be seen from the loading data in Figure 10, the peak load average has been held to an average of 5.5 - 6.0 whereas total CPU time consumption, shown in Figure 8, has continued to rise.

Several problems were noted in the loading control system that required improvements in monitor functions this past year:

- 1) Users frequently wanted to designate a job as low priority or "background" so that it would run only when the system is lightly loaded and "go to sleep" otherwise.
- 2) Scheduled demonstration jobs were receiving no advantage in performance over other jobs, other than that due to holding the load average down. A scheme was needed to cause demo jobs always to be scheduled preferentially.
- 3) Forcible control of uncooperative jobs was initially implemented by detaching them or logging them out in extreme cases. This could cause loss of important work and a less destructive yet effective mechanism was needed.
- 4) A loophole for uncooperative jobs existed that would bypass controls with good probability. If more than one user were asked to slow down at a given time, one of those jobs could refuse to cooperate and continue intensive computing while the others slowed down. Frequently, the load reduction from cooperating jobs was enough to remove the overload condition during common, local bursts of usage. Thus, with the overload gone, the uncooperative user could continue without ever having slowed down.

To improve the control system, we implemented two new scheduler control functions. First, a job can be designated to run out of a given queue no matter how much CPU time it wants to consume. This allows demo jobs always to be scheduled out of the highest priority queue assuring a better service level. It also allows background jobs to be scheduled always from the low priority queues so they only run if nothing else is to be done.

Second, a job can be stopped for a specified period of time without ever being scheduled. This function allows uncooperative jobs to be slowed for a large percentage of time (max 97.5% currently) when their load must be reduced forcibly but does not do any other damage to the operation of such jobs that could result in lost work.

These new features have substantially improved the effectiveness of the overload control system. The loophole for uncooperative jobs was plugged by noting whether jobs requested to stop make any attempt to cooperate during the assigned grace period. If there is no change in their rate of CPU time consumption, the grace period is shortened so they will be forcibly stopped before more cooperative users stop and remove the overload.

#### Other Enhancements

We have made improvements in SUMEX system software in numerous other areas including the EXECutive program, the BSYS system for file archiving and retrieving, the printer spoolers, the CHECKDSK program for verifying file system integrity, system diagnostic programs, a monitor crash analysis program, and many smaller utility extensions and bug fixes. We have updated the EXEC to be compatible with the latest version running at other TENEX sites, incorporating the extensions we have made locally. The BSYS program has been updated to the

latest version available from BBN using their system for file restoration automation. Several bugs in the improved CHECKDSK program for verifying file system integrity have been made and improvements to give users a better idea of file names that might have been lost during a crash. Improved crash and system analysis programs have been developed to assist in sorting through the complex interlinked monitor tables when unraveling a core dump to determine the cause of a crash. These include several display programs to observe the dynamic operation of individual job structures or the ARPANET. These tools have been invaluable in tracking down the difficult bugs that remain in the system.

### 2.1.2.3 Network Communication Facilities

A highly important aspect of the SUMEX system is effective communication with remote users. In addition to the economic arguments for terminal access, networking offers other advantages for shared computing. These include improved inter-user communications, more effective software sharing, uniform user access to multiple machines and special purpose resources, convenient file transfers, more effective backup, and co-processing between remote machines.

Until this past year, we have based our remote communication services on two networks - TYMNET and ARPANET. These were the only networks existing at the start of the project which allowed foreign host access. A third commercial network system, TELENET, is now competitively operational and offers a growing selection of services. During this report period we established an experimental connection to TELENET to evaluate its technical and economic advantages relative of our existing connections. The results of this experiment are reported below.

Users asked to accept a remote computer as if it were next door will use a local telephone call to the computer as a standard of comparison. Current network terminal facilities do not quite accomplish the illusion of a local call. Data loss is not a problem in most network communications - in fact with the more extensive error checking schemes, data integrity is higher than for a long distance phone link. On the other hand, networking relies upon shared community use of telephone lines to procure widespread geographical coverage at substantially reduced cost. However, unless enough total line capacity is provided to meet peak loads, substantial queueing and traffic jams result in the loss of terminal responsiveness. Limited responsiveness for character-oriented TENEX interactions continues to be a problem for network users.

#### TYMNET:

TYMNET provides broad geographic coverage for terminal access to SUMEX, spanning the country and also increasingly accessible from foreign countries (see Figure 4 on page 21). Technical aspects of our connection to TYMNET have remained unchanged this past year and have continued to operate reliably. The total use of TYMNET dropped during the TELENET experimental connection (see Figure 14) but is now increasing again since the TELENET service was dropped.

TYMNET has made few technical changes to their network that affect us other than to broaden geographical coverage. The previous network delay problems are

still apparent although better cross-country trunks into New York and New England are available improving service there. TYMNET is still primarily a terminal network designed to route users to an appropriate host and more general services such as outbound connections originated from a host or interhost connections are only done on an experimental basis. This presumably reflects the lack of current economic justification for these services among the predominantly commercial users of the network. Whereas TYMNET is developing interfaces meeting X.25 protocol standards, the internal workings of the network will likely remain the same, namely, constructing fixed logical circuits for the duration of a connection and multiplexing characters in packets over each link between network nodes from any users sharing that link as part of their logical circuit.

We have continued to purchase TYMNET services through the NLM contract with TYMNET, Inc. Because of current tariff provisions, there is no longer an economic advantage to this based on usage volume. SUMEX charges are computed on its usage volume alone and not the aggregate volume with NLM's contribution to achieve a lower rate. A new tariff provision, based on "dedicated port" pricing, is advantageous to us though. This allows purchase of a number of logical network ports at the host for a fixed cost per month, independent of connect time or number of characters transmitted. Based on previous usage data, SUMEX could save approximately \$1,000 per month in service charges by taking advantage of this charging scheme. We will continue to work closely with NIH-BRP and NLM to achieve the most cost-effective purchase of these services.

#### ARPANET:

We continue our advantageous connection to the Department of Defense's ARPANET, now managed by the Defense Communications Agency (DCA). Current ARPANET geographical and logical maps are shown in Figure 5 and Figure 6 on page 22. Consistent with agreements with ARPA and DCA we are enforcing a policy that restricts the use of ARPANET to users who have affiliations with DoD-supported contractors and system/software interchange with cooperating network sites. We have maintained good working relationships with other sites on the ARPANET for system backup and software interchange. Such day-to-day working interactions with remote facilities would not be possible without the integrated file transfer, communication, and terminal handling capabilities unique to the ARPANET. The ARPANET is also key to maintaining on-going intellectual contacts between SUMEX projects such as the Stanford Heuristic Programming Project authorized to use the net and other active AI research groups in the ARPANET community.

#### TELENET

We recognize the importance of effective, economical communication facilities for SUMEX-AIM users and are continuously looking for ways to improve our existing facilities. During the past year, based on the approval of the AIM Executive Committee and the NIH-BRP, we established an experimental connection to the TELENET network to evaluate its performance for support of the SUMEX-AIM community (see Figure 7 on page 24 for an illustration of the current geographic coverage of TELENET). Our connection was via a TP-2200 interface with 12 asynchronous lines to the SUMEX host and one 4800 baud line connecting to the network proper. TELENET has many attractive features in terms of a symmetry analogous to that of the ARPANET for terminal traffic and file transfers and

being a commercial network, it does not have the access restrictions of the ARPANET. Its tariff schedule also affords lower costs than TYMNET for comparable service volume.

However, despite system changes we made to optimize TELENET performance (Xon/Xoff facilities to improve traffic flow), users felt a substantial degradation in service when using TELENET as opposed to TYMNET. We insisted that users use TELENET whenever possible between November 1978 and May 1979 to maximize user accommodation so that problems arising from differences in access conventions would not cloud judgements of services. Complaints included poor node reliability, intolerable delays in response, uneven flow of terminal output, and poor operational management of the network in keeping users informed of network and host status. From the system viewpoint at SUMEX, we detected similar problems. We received ineffective system engineering support in trying to tune network parameters to optimize performance for our user community and poor or erroneous feedback about network failures and problem resolution. In practice, TELENET offered no service advantages over TYMNET, since no file transfer connections above 1200 baud are currently allowed, no facilities to control local versus remote echoing exist, and no electronic mail system exists to facilitate communication between network operations staff and host nodes. Also company financial problems portend substantial delays in remedying these problems.

Because of grant budget limitations, we were forced to decide between the TYMNET and TELENET connections - only one could be afforded. Based on the distinct user preference expressed for TYMNET, we decided to terminate the TELENET connection as of May 1, 1979. We will continue to monitor TELENET developments (and those of other potential national network servers, e.g., AT&T, IBM, and Xerox) and may recommend a reevaluation of an alternative source for network services in the future.

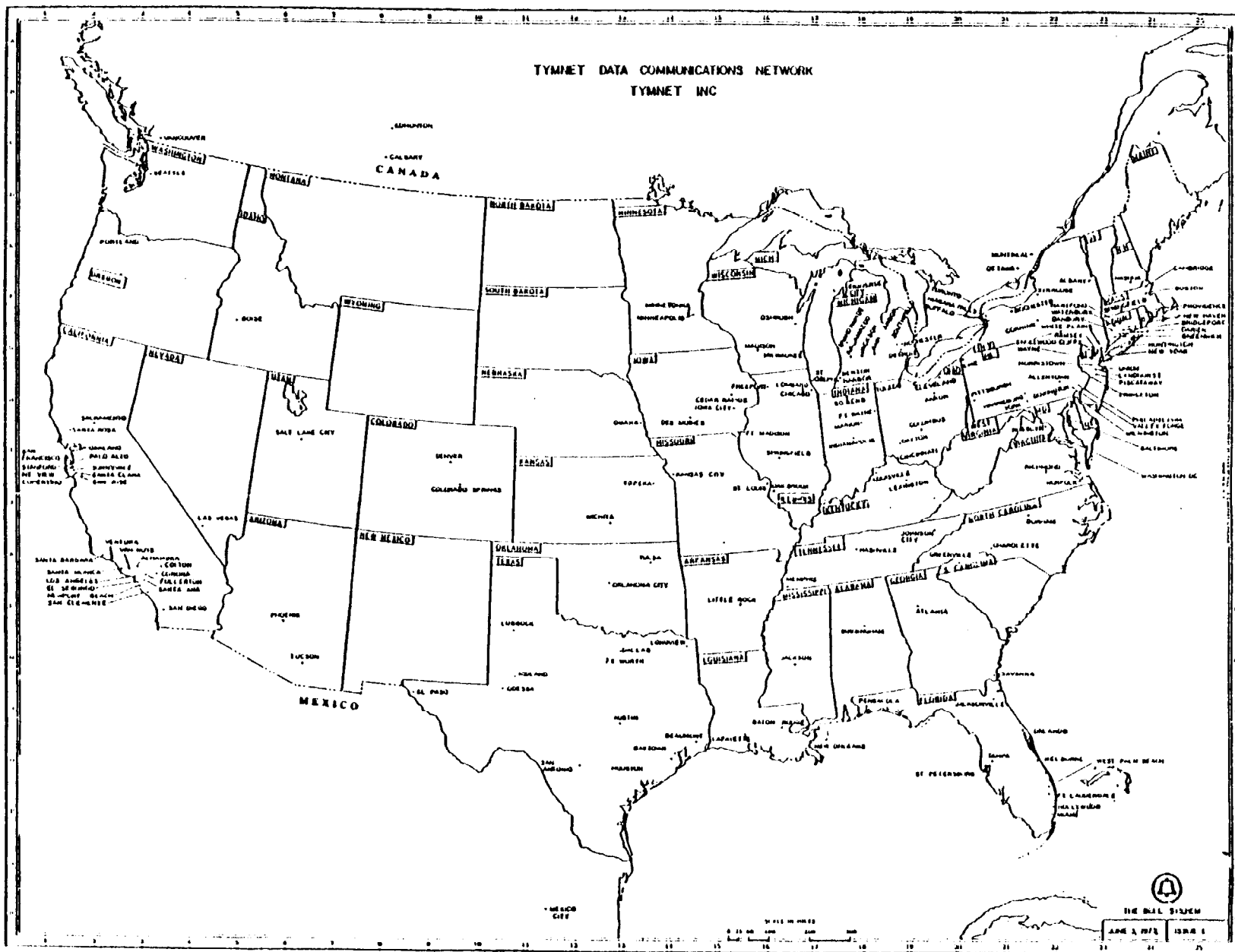


Figure 4. TYMNET Network Map

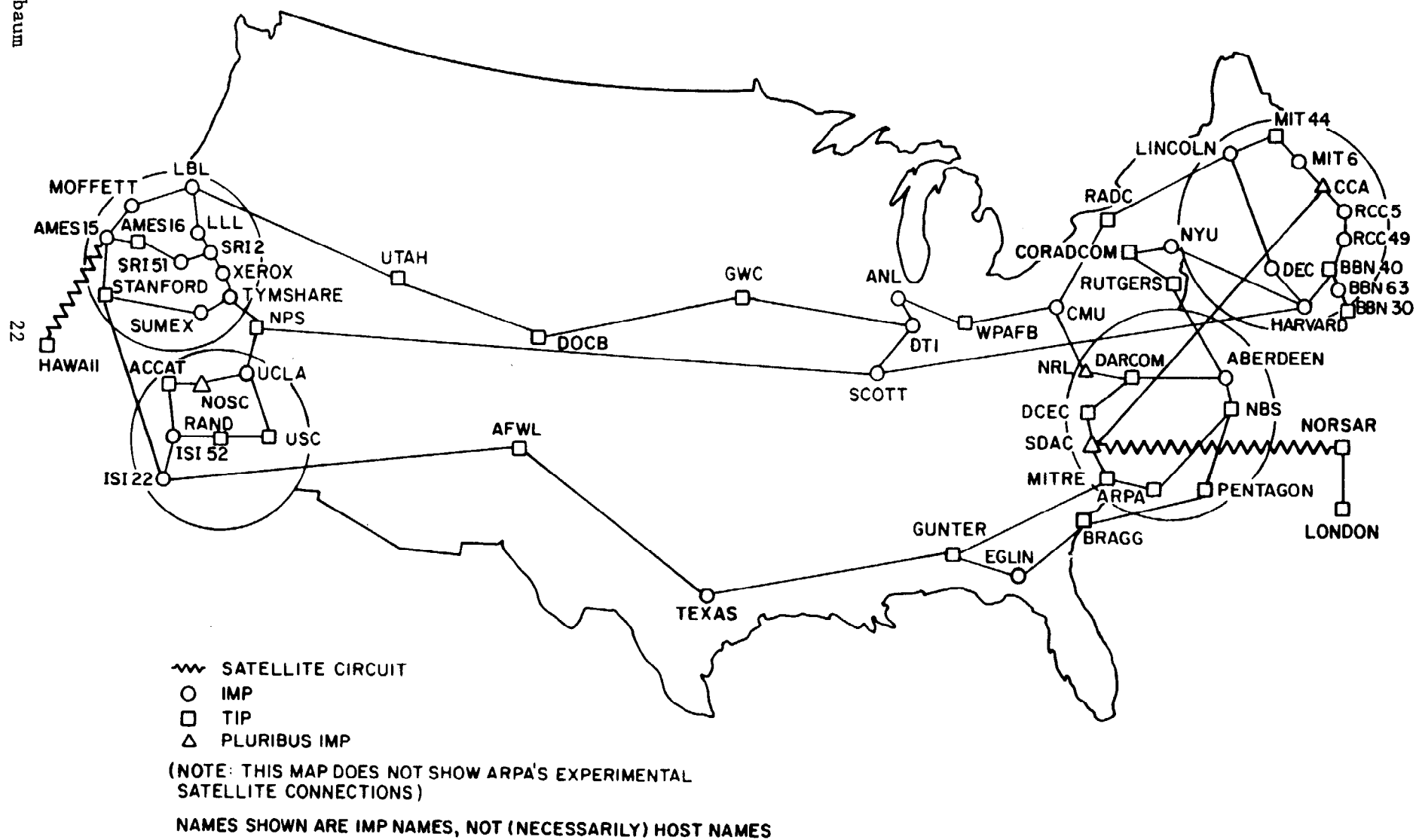
Figure 5. **ARPANET GEOGRAPHIC MAP, MARCH 1979**

Figure 6. ARPANET LOGICAL MAP, MARCH 1979

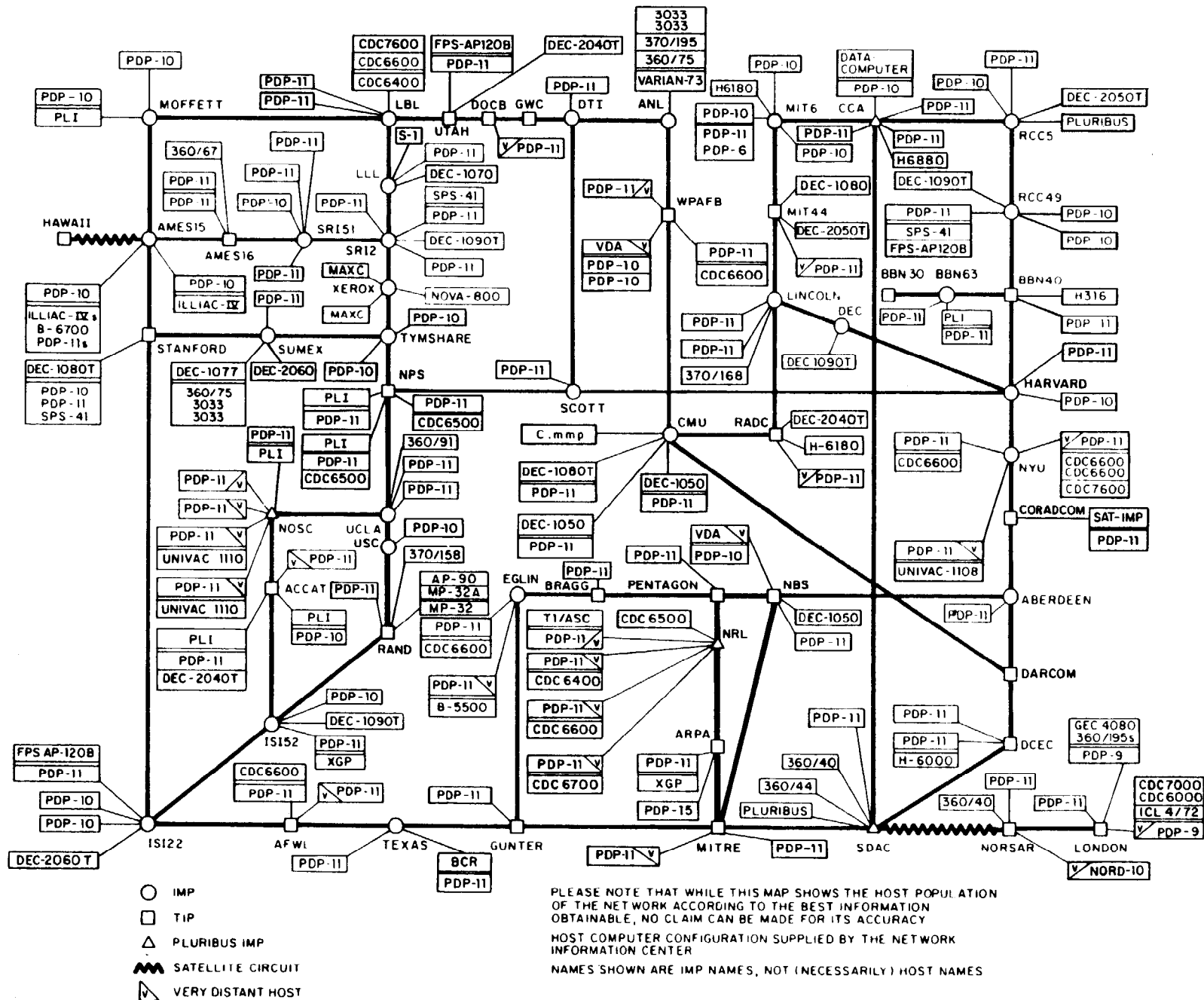
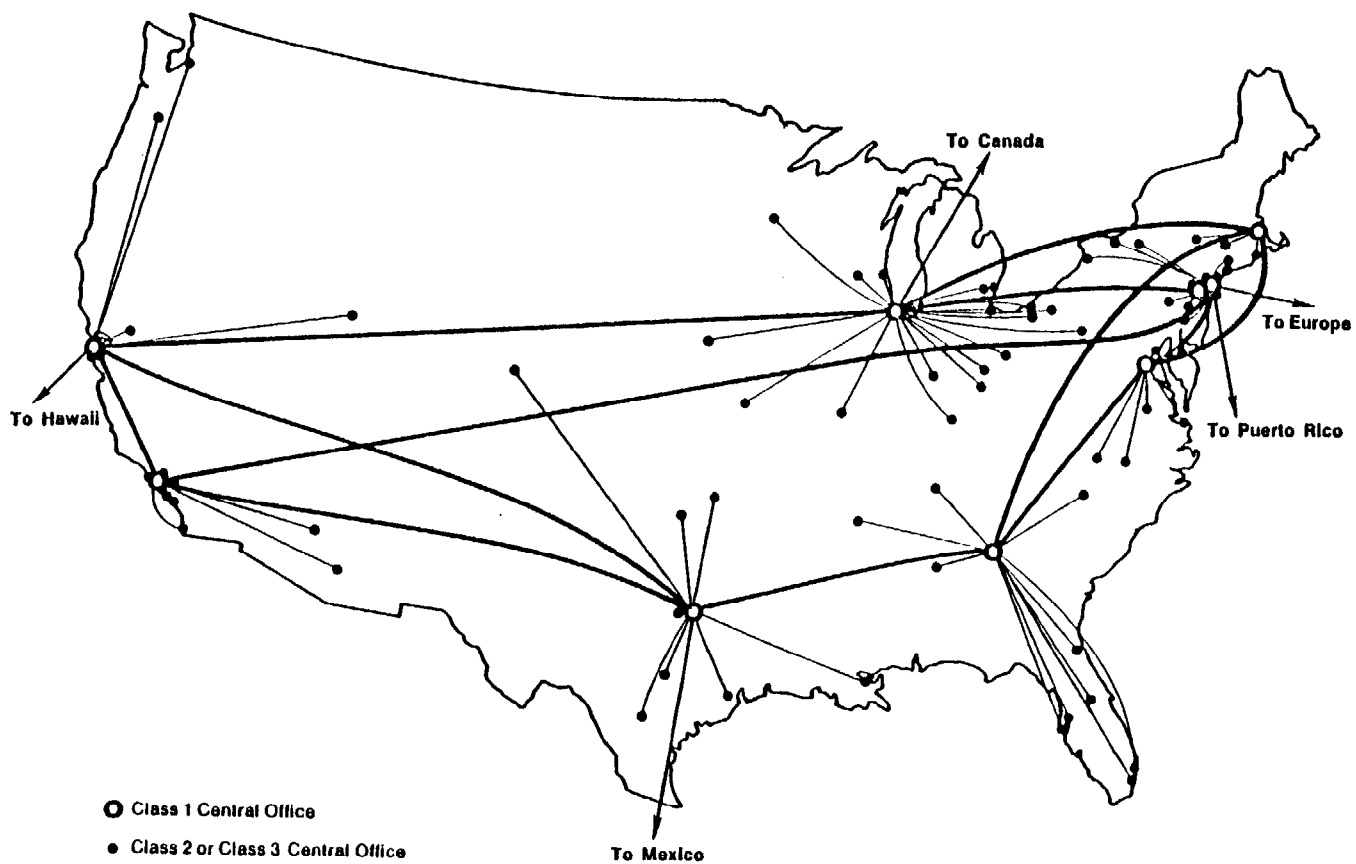
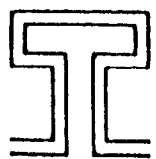


Figure 7

# THE TELENET NETWORK



2.1.2.4 System Reliability and Backup

System reliability has been very good on average with several periods of particular hardware or software problems. The table below shows monthly system reloads and downtime for the past year. It should be noted that the number of system reloads is greater than the actual number of system crashes since two or more reloads may have to be done within minutes of each other after a crash to repair file damage or to diagnose the cause of failure.

	1978						1979					
	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APR
<u>RELOADS</u>												
Hardware	6	8	5	6	8	10	1	4	2	2	6	4
Software	0	0	4	5	9	9	5	3	9	4	7	10
Environmental	3	0	1	0	1	0	0	0	1	0	1	0
Unknown Cause	7	4	1	4	5	5	1	1	0	0	1	1
	--	--	--	--	--	--	--	--	--	--	--	--
Totals	16	12	11	15	23	24	7	8	12	6	15	15
<u>DOWNTIME (Hrs)</u>												
Unscheduled	36	22	33	37	28	37	3	14	8	16	17	14
Scheduled	38	34	22	25	20	31	30	20	22	17	33	16
	--	--	--	--	--	--	--	--	--	--	--	--
Totals (Hrs)	74	56	55	62	48	68	33	34	30	33	50	30

TABLE 1. System Reliability by Month

During the year, we encountered several hardware problems that caused temporary increases in the number of crashes. These were very intermittent problems that were difficult to isolate and account for the increased number of reloads during September and October 1978 and again in March and April of 1979. Several problems resulted from oxidation of electrical contacts and we might expect an increase in such age-related failures as the system gets older.

Probably the most serious hardware failure was a head crash on one of the swapping disks. A rubber diaphragm burst forcing one set of heads to contact a platter. The debris from that crash then spread to the other surfaces and caused those heads to crash. We expect repairs to be complete by early July. This may forecast other problems caused by aging of rubber parts in the swapping disks and we will take steps to replace these if need be before another failure results.

We have had an on-going effort to increase software reliability and have fixed a number of bugs that have been perennial causes of crashes or file loss at system shut-down. Some of these fixes have required setting system stops to get appropriate dumps to analyze the problem causes and thereby also temporarily increased the number of crashes.